Bedrock Computer Technologies, LLC v. Softlayer Technologies, Inc. et al Doc. 842 Att. 23

EXHIBIT 22

Dockets.Justia.com

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION

BEDROCK COMPUTER TECHNOLOGIES LLC,	§	
	§	
Plaintiff,	§	
	§	**CASE NO. 6:09-cv-269-LED**
v.	§	
	§	**Jury Trial Demanded**
SOFTLAYER TECHNOLOGIES, INC., et al.	§	
	§	
Defendants.	§	

PLAINTIFF BEDROCK COMPUTER TECHNOLOGIES LLC'S OBJECTIONS AND RESPONSES TO YAHOO! INC.'S THIRD SET OF INTERROGATORIES (NOS. 4-7)

TO: Defendant Yahoo! Inc., by and through its attorney of record, Yar R. Chaikovsky, MCDERMOTT WILL & EMERY LLP, 275 Middlefield Road, Suite 100, Menlo Park, California 94025.

Pursuant to Federal Rules of Civil Procedure 33, Plaintiff Bedrock Computer Technologies LLC ("Plaintiff" or "Bedrock") provides the following objections and responses to Yahoo! Inc.'s ("Yahoo") Third Set of Interrogatories (Nos. 4-7).

PRELIMINARY STATEMENT

1. Bedrock incorporates by reference each and every general objection set forth below into each specific response. The failure to include any general objection in any specific response shall not be interpreted as a waiver of any general objection to that response.

2. By responding to Yahoo's interrogatories, Bedrock does not waive any objection that may be applicable to: (a) the use, for any purpose, by Yahoo of any information or

documents given in response to Yahoo's interrogatories; or (b) the admissibility, relevance, or materiality of any of the information or documents to any issue in this case.

3. No incidental or implied admissions are intended by the responses herein. The fact that Bedrock has answered or objected to any interrogatory should not be taken as an admission that Bedrock accepts or admits the existence of any "fact" set forth or assumed by such interrogatory.

4. Bedrock's responses to Yahoo's interrogatories are made to the best of Bedrock's present knowledge, information, and belief. Bedrock reserves the right to supplement and amend these responses should future investigation indicate that such supplementation or amendment is necessary. Bedrock reserves the right to make any use of, or introduce at any hearing or trial, information or documents that are responsive to Yahoo's interrogatories, but discovered subsequent to Bedrock's service of these responses, including, but not limited to, any information or documents obtained in discovery herein.

GENERAL OBJECTIONS

1. Bedrock objects to each interrogatory to the extent that it seeks information already in Yahoo's possession, a matter of public record or otherwise equally available to any Defendant.

2. Bedrock objects to each interrogatory to the extent that it seeks the identification of "all," "every," "any," and "each" entity, person, or document that refers to a particular subject. Bedrock will comply with the Federal Rules and the Local Rules and will use reasonable diligence to identify responsive persons or documents.

3. Bedrock's responses herein, and its disclosure of information pursuant to these responses, do not in any way constitute an adoption of Yahoo's purported definitions of words

and/or phrases contained in Yahoo's interrogatories. Bedrock objects to these definitions to the extent that they: (a) are unclear, vague, overly broad, or unduly burdensome; (b) are inconsistent with the ordinary and customary meaning of the words or phrases they purport to define; (c) include assertions of purported fact that are inaccurate or at the very least disputed by the parties to this action; and/or (d) incorporate other purported definitions that suffer from such defects.

4. Bedrock objects to each and every interrogatory to the extent that it purports, through Yahoo's definitions, instructions to the extent that they are inconsistent with, or not authorized by, the Federal Rules of Civil Procedure, the Local Rules of the Eastern District of Texas, or the Court's Patent Rules and discovery orders.

5. Bedrock objects to the extent that the interrogatories call for information protected by the attorney-client privilege, the attorney work product doctrine or any other applicable doctrine, privilege or immunity. Any disclosure of privileged information is inadvertent and should be deemed to have no legal effect or consequence, and Bedrock does not waive any privilege upon such inadvertent disclosure.

6. Bedrock objects to each and every interrogatory to the extent that it seeks information that is cumulative or duplicative of information, disclosures, or discovery already provided by Bedrock.

7. Bedrock objects to the inclusion of "Bedrock's affiliates, parents, divisions, joint ventures, assigns, predecessors and successors in interest" and "former employees, counsel, agents, consultants, representatives, and any other person acting on behalf of the foregoing" in the definitions of "Bedrock," "you," "your," and "plaintiff" to the extent that the interrogatories using these definitions are requesting information that is not in the possession, custody, or

control of Bedrock or seeking information that is protected by a doctrine, privilege, or immunity from discovery.

8. Bedrock objects to Yahoo's definitions of "reflect," "reflecting," "refers to," relating to," "referring to," "identify," "identity," "identity," and "identity," on the grounds that they are vague, ambiguous, overly broad, and as used in the interrogatories, make the interrogatories unduly burdensome.

9. Bedrock objects to the Definitions of "identify," and related terms and "relates to," and related terms to the extent that they purport to require Bedrock to take action or to provide information not required by, or which exceeds the scope of, the Federal Rules of Civil Procedure.

10. Bedrock objects to the extent that the interrogatories seek information of third parties with whom Bedrock may have entered into non-disclosure or confidentiality agreements or other agreements having privacy, confidentiality, or non-disclosure provisions, which prohibit the disclosure by Bedrock of the third party's information.

11. Bedrock objects to providing responses to each interrogatory where the requested information may be derived or ascertained from documents that have been or are being produced.

12. Bedrock objects to each and every interrogatory to the extent that it seeks information that is properly the subject of expert testimony in advance of the Federal Rules of Civil Procedure, the Local Rules of the Eastern District of Texas, the Court's Patent Rules and discovery orders, or the parties' discovery stipulations.

13. Bedrock objects to the extent the interrogatories seek information that is not relevant to any claim or defense in this case, is not reasonably calculated to lead to the discovery of admissible evidence, or is otherwise not discoverable under Fed. R. Civ. P. 26(a).

14. Bedrock notifies the Defendants that it will object to interrogatories containing multiple subparts that together exceed the total number of interrogatories that the Defendants are allowed to propound pursuant to an order of the Court or the Federal Rules of Civil Procedure. For purposes of this objection, Bedrock will count interrogatory subparts as part of one interrogatory for the purpose of numerically limiting interrogatories to the extent that such subparts are logically or factually subsumed within and necessarily related to the primary question. To the extent any subsequent question can stand alone or is independent of the first question, such subsequent question is a discrete interrogatory. Accordingly, Bedrock will count discrete or separate questions as separate interrogatories, notwithstanding they are joined by a conjunctive word and may be related. Bedrock will endeavor, however, to treat genuine subparts as subparts and will not count such genuine subparts as separate interrogatories. For purposes of this objection, a subpart inquiring on the same topic as the interrogatory therefore will not itself qualify as a separately counted interrogatory, but when the interrogatory subpart introduces a new topic that is in a distinct field of inquiry, the subpart then assumes separate interrogatory status for the purpose of counting. *See Orion IP, LLC v. Staples, Inc., et al.*, No. 2:04-CV-297, at *1 (E.D. Tex July 7, 2005) (Dkt. No. 171).

OBJECTIONS AND RESPONSES TO SPECIFIC INTERROGATORIES

INTERROGATORY NO. 4:

For each claim of the ’120 Patent, describe all investigations made by or on behalf of Bedrock, Richard Nemes, Mikhail Lotvin, or David Garrod prior to filing of the Complaint regarding whether any claim of the ’120 Patent is infringed by Yahoo! product or service, including identifying the persons involved in the investigations, the persons to whom reports were made, the persons involved in the approval of the filing of the Complaint, the date of the investigation, the Yahoo! products and services that were the subject of the investigation, the public information considered in the investigation, any other items or information considered in the investigation, when and where such information and items were obtained, the conclusions reached in the investigations, all documents referring to or describing such investigations, and the date on which Bedrock, Richard Nemes, Mikhail Lotvin, or David Garrod first became aware that any of Yahoo!’s accused products or services might infringe the ’120 Patent.

ANSWER:

In addition to the general objections, Bedrock specifically objects to this interrogatory to the extent that it seeks the production, identification, or disclosure of information protected by the attorney-client privilege, work product doctrine, or any other applicable privilege or doctrine. Bedrock further objects that the information to the extent that it seeks the disclosure of information that is properly the subject of expert testimony; such information will be disclosed consistent with the Court’s Docket Control Order and the deadline for burden expert reports.

Subject to the foregoing specific and general objections, Bedrock responds as follows. Bedrock became aware of the infringement of software based on the publicly available Linux

kernel through inspection of the publicly available Linux kernel prior to filing suit. As such, Bedrock incorporates by reference its infringement contentions that it served on Google on October 9, 2009. Bedrock became aware that Yahoo operates software based on the publicly available Linux kernel through the website http://www.tribbleagency.com/?m=200811.

INTERROGATORY NO. 5:

For each asserted claim of the '120 Patent, explain the basis for any contention by Bedrock that the claim, if implemented, prevents or in any way hinders a denial of service attack against servers implementing the claim, the specific claim element(s) allegedly preventing or hindering such an attack, and including any evidence which forms the basis for any such contention. Your answer should include the witnesses upon which Bedrock relies to support this contention, their anticipated testimony, and the specific portions of the documents or other information upon which the witnesses or Bedrock relies.

ANSWER:

In addition to the general objections, Bedrock specifically objects to this interrogatory to the extent that it seeks the production, identification, or disclosure of information protected by the attorney-client privilege, work product doctrine, or any other applicable privilege or doctrine. Bedrock further objects that the information to the extent that it seeks the disclosure of information that is properly the subject of expert testimony; such information will be disclosed consistent with the Court's Docket Control Order and the deadline for burden expert reports.

Subject to the foregoing general and specific objections, Bedrock responds as follows. All claims of the '120 Patent require the identification and removal of expired records when a linked list is accessed by a record searching routine. *See* '120 Patent at claims 1-8. "This incremental garbage collection technique has the decided advantage of automatically eliminating

unneeded records without requiring that the information system be taken off-line for such garbage collection." *See* '120::2:64-67. In a system that does not implement the on-the-fly garbage collection techniques claimed in the '120 Patent, expired records "will, in time, seriously degrade the performance of the retrieval system." *See* '120::5:41-52. "Degradation shows up in two ways. First, the presence of expired records lengthens search times since they cause the external chains to be longer than they otherwise would be. Second, expired records occupy dynamically allocated memory storage that could be returned to the system memory for useful allocation." *See id.*

A denial of service attacker could exploit these disclosed weaknesses of a system not implementing the claims of the '120 Patent. In fact, the very weaknesses described in the '120 Patent were identified as weaknesses in pre-infringing versions of Linux. Specifically, on-the-fly garbage collection was added to the routine rt_intern_hash of the Linux routing cache code in the summer of 2003 to solve a serious performance problem of the Linux routing cache when under heavy load. Bedrock hereby incorporates by reference its infringement contentions. The goal of the change was to cut down on the frequency and duration of conventional garbage collections in the Linux routing cache. Under heavy TCP/IP loads, users were seeing the Linux boxes performing poorly and dropping TCP/IP packets. The problem was traced back to the routing cache garbage collector keeping the routing cache locked for long periods of time when there is heavy TCP/IP traffic.

The routing cache performance problem is memorialized in a Linux-net email list of Linux users and Linux network stack developers. The message thread was entitled "Route cache performance under stress." For example, a user named "CIT/Paul" described the problem as follows:

> Try forwarding packets generated by juno-z.101f.c and it adds EVERY packet to the route cache.. Every one. And at 30,000 pps It destroys the cache because every single packet coming in is NOT in the route cache because it's random ips. Nothing you can do About that except make the cache and everthing related to it wicked faster, OR remove the per packet additions to the cache.

See BTEX0748686-88 (http://marc.info/?l=linux-net&m=105513656320859&w=2).

The on-the-fly garbage collection solution to the routing cache performance problems was first proposed by David Miller of Red Hat in a number of email messages on the same message thread:

> Here is a simple idea, make the routing cache miss case steal an entry sitting at the end of the hash chain this new one will map to. It only steals entries which have not been recently used.

See BTEX0748689 (http://marc.info/?l=linux-net&m=105513880722053&w=2).

> My main current quick idea is to make rt_intern_hash() attempt to flush out entries in the same hash chain instead of allocating new entries.

See BTEX0748690 (http://marc.info/?l=linux-net&m=105514015222804&w=2).

> We have to walk the entire destination hash chain _ANYWAYS_ to verify that a matching entry has not been put into the cache while we were procuring the new one. During this walk we can also choose a candidate rtcache entry to free.

See BTEX0748691-92 (http://marc.info/?l=linux-net&m=105514201423926&w=2).

> The problem is that GC cannot currently keep up with DoS like traffic pattern. As a result, routing latency is not smooth at all, you get spikes because each GC run goes for up to an entire jiffie because it has so much work to do. Meanwhile, during this expensive GC processing, packet processing is frozen on UP system.

See BTEX0748693-94 (http://marc.info/?l=linux-net&m=105514238324129&w=2). In this way, the Linux community in 2003 identified that the standalone garbage collection routine in Linux was the essentially taking servers off-line, just as described by Dr. Nemes in the ’120 Patent in 1999.

On-the-fly garbage collection was first added to rt_intern_hash in Linux kernel version 2.5.72 in June 2003 by David Miller and Alexey Kuznetsov, another Linux networking developer. The changes became a permanent part of the Linux kernel going forward.

The original message thread on the Linux routing cache was begun in the April 2003 when a security researcher by the name of Florian Weimer first identified another serious performance problem in the Linux routing cache related to how the routing cache code was doing hashing. Mr. Weimer described this problem:

> Please read the following paper:
>
> <http://www.cs.rice.edu/~scrosby/tr/HashAttack.pdf>
>
> Then look at the 2.4 route cache implementation.
>
> Short summary: It is possible to freeze machines with 1 GB of RAM and more with a stream of 400 packets per second with carefully chosen source addresses. Not good.

See BTEX0748695-96 (http://marc.info/?l=linux-kernel&m=104956079213417&w=2). This type of denial of service attack, which is called an algorithmic complexity attack, works by intentionally lengthening a linked list. In this way, the Linux community identified in 2003 that the lengthening of a linked list, as described by Dr. Nemes in the '120 Patent in 1999, causes serious system performance degradation. On-the-fly garbage removal is a solution to this type of denial of service attack as well, but Linux developers, even when faced with this problem, did not come up with on-the-fly garbage collection as the solution.

INTERROGATORY NO. 6:

For each Accused Instrumentality identified in Bedrock's Infringement Contentions, describe in detail the basis for any contention by Bedrock that the module/net/ipv4/route.c prevents or in any way hinders a denial of service attack against servers implementing this module, including any evidence which forms the basis for any such contention. Your answer should include the witnesses upon which Bedrock relies to support this contention, their

anticipated testimony, and the specific portions of the documents or other information upon which the witnesses or Bedrock relies.

ANSWER:

In addition to the general objections, Bedrock specifically objects to this interrogatory to the extent that it seeks the production, identification, or disclosure of information protected by the attorney-client privilege, work product doctrine, or any other applicable privilege or doctrine. Bedrock further objects that the information to the extent that it seeks the disclosure of information that is properly the subject of expert testimony; such information will be disclosed consistent with the Court's Docket Control Order and the deadline for burden expert reports.

Subject to the foregoing general and specific objections, Bedrock responds as follows. All claims of the '120 Patent require the identification and removal of expired records when a linked list is accessed by a record searching routine. *See* '120 Patent at claims 1-8. "This incremental garbage collection technique has the decided advantage of automatically eliminating unneeded records without requiring that the information system be taken off-line for such garbage collection." *See* '120::2:64-67. In a system that does not implement the on-the-fly garbage collection techniques claimed in the '120 Patent, expired records "will, in time, seriously degrade the performance of the retrieval system." *See* '120::5:41-52. "Degradation shows up in two ways. First, the presence of expired records lengthens search times since they cause the external chains to be longer than they otherwise would be. Second, expired records occupy dynamically allocated memory storage that could be returned to the system memory for useful allocation." *See id.*

A denial of service attacker could exploit these disclosed weaknesses of a system not implementing the claims of the '120 Patent. In fact, the very weaknesses described in the '120

patent were identified as weaknesses in pre-infringing versions of Linux. Specifically, on-the-fly garbage collection was added to the routine rt_intern_hash of the Linux routing cache code in the summer of 2003 to solve a serious performance problem of the Linux routing cache when under heavy load. Bedrock hereby incorporates by reference its infringement contentions. The goal of the change was to cut down on the frequency and duration of conventional garbage collections in the Linux routing cache. Under heavy TCP/IP loads, users were seeing the Linux boxes performing poorly and dropping TCP/IP packets. The problem was traced back to the routing cache garbage collector keeping the routing cache locked for long periods of time when there is heavy TCP/IP traffic.

The routing cache performance problem is memorialized in a Linux-net email list of Linux users and Linux network stack developers. The message thread was entitled "Route cache performance under stress." For example, a user named "CIT/Paul" described the problem as follows:

> Try forwarding packets generated by juno-z.101f.c and it adds EVERY packet to the route cache.. Every one. And at 30,000 pps It destroys the cache because every single packet coming in is NOT in the route cache because it's random ips. Nothing you can do About that except make the cache and everthing related to it wicked faster, OR remove the per packet additions to the cache.

See BTEX0748686-88 (http://marc.info/?l=linux-net&m=105513656320859&w=2).

The on-the-fly garbage collection solution to the routing cache performance problems was first proposed by David Miller of Red Hat in a number of email messages on the same message thread:

> Here is a simple idea, make the routing cache miss case steal an entry sitting at the end of the hash chain this new one will map to. It only steals entries which have not been recently used.

See BTEX0748689 (http://marc.info/?l=linux-net&m=105513880722053&w=2).

> My main current quick idea is to make rt_intern_hash() attempt to flush out entries in the same hash chain instead of allocating new entries.

See BTEX0748690 (http://marc.info/?l=linux-net&m=105514015222804&w=2).

> We have to walk the entire destination hash chain _ANYWAYS_ to verify that a matching entry has not been put into the cache while we were procuring the new one. During this walk we can also choose a candidate rtcache entry to free.

See BTEX0748691-92 (http://marc.info/?l=linux-net&m=105514201423926&w=2).

> The problem is that GC cannot currently keep up with DoS like traffic pattern. As a result, routing latency is not smooth at all, you get spikes because each GC run goes for up to an entire jiffie because it has so much work to do. Meanwhile, during this expensive GC processing, packet processing is frozen on UP system.

See BTEX0748693-94 (http://marc.info/?l=linux-net&m=105514238324129&w=2). In this way, the Linux community in 2003 identified that the standalone garbage collection routine in Linux was the essentially taking servers off-line, just as described by Dr. Nemes in the '120 Patent in 1999.

On-the-fly garbage collection was first added to rt_intern_hash in Linux kernel version 2.5.72 in June 2003 by David Miller and Alexey Kuznetsov, another Linux networking developer. The changes became a permanent part of the Linux kernel going forward.

The original message thread on the Linux routing cache was begun in the April 2003 when a security researcher by the name of Florian Weimer first identified another serious performance problem in the Linux routing cache related to how the routing cache code was doing hashing. Mr. Weimer described this problem:

> Please read the following paper:
>
> <http://www.cs.rice.edu/~scrosby/tr/HashAttack.pdf>
>
> Then look at the 2.4 route cache implementation.

> Short summary: It is possible to freeze machines with 1 GB of RAM and more with a stream of 400 packets per second with carefully chosen source addresses. Not good.

See BTEX0748695-96 (http://marc.info/?l=linux-kernel&m=104956079213417&w=2). This type of denial of service attack, which is called an algorithmic complexity attack, works by intentionally lengthening a linked list. In this way, the Linux community identified in 2003 that the lengthening of a linked list, as described by Dr. Nemes in the '120 Patent in 1999, causes serious system performance degradation. On-the-fly garbage removal is a solution to this type of denial of service attack as well, but Linux developers, even when faced with this problem, did not come up with on-the-fly garbage collection as the solution.

INTERROGATORY NO. 7:

For each Accused Instrumentality identified in Bedrock's Infringement Contentions describe in detail the basis for any contention by Bedrock that module /net/ipv5/route.c and the code identified by Bedrock in its Infringement Contentions and/or its response to Yahoo!'s Interrogatory No. 3 as infringing the '120 patent [*sic*] are the basis for customer demand for such Accused Instrumentality, including any evidence which forms the basis for any such contention. Your answer should include the witnesses upon which Bedrock relies to support this contention, their anticipated testimony, and the specific portions of the documents or other information upon which the witnesses or Bedrock relies.

ANSWER:

In addition to the general objections, Bedrock specifically objects to this interrogatory to the extent that it seeks the production, identification, or disclosure of information protected by the attorney-client privilege, work product doctrine, or any other applicable privilege or doctrine. Bedrock further objects to this interrogatory to the extent that it seeks the disclosure of information that is properly the subject of expert testimony; such information will be disclosed

consistent with the Court's Docket Control Order and the deadline for burden expert reports. Furthermore, the discovery in this case is on-going. Bedrock's efforts to discover all facts related to this interrogatory have been hindered by the Defendants' refusal to respond to Bedrock's discovery requests. In fact, Bedrock has been required to move the Court for the discovery due to the Defendants continued failures to respond.

Subject to the foregoing general and specific objections, Bedrock responds as follows:

The internet has become a powerful and paradigm-changing tool for the United States and World economies. Businesses have become dependant on fast, reliable, always on services provided by websites. The Defendants, and their customers, rely upon infringing versions of Linux to provide the fast, reliable, and always-on services. Denial of service attacks directly challenge a company's ability to provide these fast, reliable, and always on services to their customers. A final determination as to whether the entire market value rule applies in this case will be made by Bedrock's expert. However, the Federal Circuit has made it clear that, even if the entire market value rule is not appropriate, "the base used in a running royalty calculation can *always* be the value of the entire commercial embodiment so long as the rate is within an acceptable range (as determined by the evidence)." *Lucent Technologies, Inc. v. Gateway Inc.*, 580 F.3d 1301 at 1338-39 (Fed. Cir. 2009)(emphasis added). Bedrock is still investigating this issue and reserves the right to supplement. By way of example, but in no way limited to the documents specifically identified herein, Bedrock has produced relevant, responsive documents, from which a response to this interrogatory may be derived or ascertained, including the following:

BTEX0748697-699 BTEX0748700-730

BTEX0748731-750 BTEX0748751-52

Date: September 2, 2010.

Respectfully submitted,

/s/ Jason D. Cassady

Sam F. Baxter
Texas Bar No. 01938000
McKOOL SMITH, P.C.
Email: sbaxter@mckoolsmith.com
104 E. Houston Street, Suite 300
Marshall, Texas 75670
Telephone: (903) 923-9000
Facsimile: (903) 923-9099

Douglas A. Cawley, Lead Attorney
Texas Bar No. 04035500
Email: dcawley@mckoolsmith.com
Theodore Stevenson, III
Texas Bar No. 19196650
Email: tstevenson@mckoolsmith.com
Jason D. Cassady
Texas State Bar No. 24045625
Email: jcassady@mckoolsmith.com
J. Austin Curry
Texas Bar No. 24059636
Email: acurry@mckoolsmith.com
McKOOL SMITH, P.C.
300 Crescent Court, Suite 1500
Dallas, Texas 75201
Telephone: 214-978-4000
Facsimile: 214-978-4044

Robert M. Parker
Texas Bar No. 15498000
E-mail: rmparker@pbatyler.com
Robert Christopher Bunt
Texas Bar No. 00787165
E-mail: rcbunt@pbatyler.com
PARKER, BUNT & AINSWORTH, P.C.
100 E. Ferguson, Suite 1114
Tyler, Texas 75702
Telephone: 903-531-3535
Facsimile: 903-533-9687

ATTORNEYS FOR PLAINTIFF BEDROCK COMPUTER TECHNOLOGIES LLC

CERTIFICATE OF SERVICE

The undersigned certifies that a true and correct copy of the foregoing document was served on counsel of record via email on September 2, 2010.

/s/ Jason D. Cassady
Jason D. Cassady